UFR Sciences et Techniques

Licence S&T – 1°année

LOGIQUE POUR L'INFORMATIQUE

Projet compétitif Prolog : traducteur français schtroumpf

1. Introduction : projet compétitif

Le présent projet a pour objectif de vous faire développer un vrai programme Prolog mettant en jeu une structure de données récursive que nous n'avons pas étudiée en cours, mais qui sera au centre de votre enseignement d'algorithmique avancée de second semestre : les **listes**. Vous aurez donc à développer les compétences acquises ces dernières semaines en termes de programmation récursive, mais également à aller chercher de vous-même des connaissances nouvelles pour comprendre la programmation des listes en Prolog.

Chaque projet reçu sera noté sur un barème de 20 points qui tiendra compte de la correction sémantique du code produit (12 points), mais également de son efficacité, sa lisibilité et de la clarté des commentaires associés (8 points). La note obtenue entrera dans la moyenne de l'UE pour ¼ de la note finale (si l'étudiant a composé dans toutes les épreuves) mais ne pourra être au préjudice de l'étudiant. La note finale de l'UE sera donc calculée comme suit :

- NF = Sup (3/4 contrôles + ¼ Projet Prolog , contrôles)

La participation à ce projet n'est pas obligatoire, sauf pour les étudiants qui ont une absence justifiée dans un des grands contrôles de l'UE à savoir :

- Ptissem Kholti
- Mostefai Mehdi
- Gay-Bellile Benoit

Ce projet se traduit par ailleurs par un aspect compétitif. En effet, les premiers étudiants parvenant à me transmettre un projet fonctionnant se verront accorder une bonification sur leur note de logique respectant le barème suivant :

- 1er projet correct reçu + 1 point sur la note de l'UE de logique
- 2ème projet reçu + 0,5 points
- 3ème projet reçu + 0,3 points
- 4ème projet reçu + 0,2 points
- 5° projet reçu + 0,1 points
- Autre projets aucune bonification

Que le meilleur … ou la meilleure gagne !

2. Une structure de donnée récursive : les listes

Le présent projet a pour objectif de vous faire développer un vrai programme Prolog mettant en jeu une structure de données récursive que nous n'avons pas étudiée en cours, mais qui sera au centre de votre enseignement d'algorithmique avancée de second semestre : les **listes**. Vous aurez donc à développer les compétences acquises ces dernières semaines en termes de programmation récursive, mais également à aller chercher de vous-même des connaissances nouvelles pour comprendre la programmation des listes en Prolog.

Les listes sont une structure de données très générale qui permet de représenter par exemple des ensembles dont le cardinal (le nombre d'éléments) n'est pas connu a priori … et peu évoluer au fil du temps. Pour permettre cela, cette structure de données se définit d'une manière récursive, suivant les règles suivantes :

- L'atome [] est une liste vide, qui représente donc un ensemble vide (c'est le cas d'arrêt)
- Une liste quelconque L se décompose en une tête T, qui représente le premier élément de la liste, et une liste Q, appelée queue de la liste. En Prolog, on notera L sous la forme [T | Q].

On voit que la définition est récursive, puisque le concept de liste est défini à partir du concept de liste lui-même (pour la queue). La queue de la liste ayant un élément en moins (T) que la liste elle-même, par récursivité, on arrive à la fin à une liste donc la queue est une liste vide : cas d'arrêt.

Ces structures sont utiles pour représenter par exemple les données langagières. Prenons une phrase donnée, on ne sait pas quelle est sa longueur, mais elle est formée d'un ensemble de mots : chaque mot sera un élément de la liste, cette structure récursive permettant de modéliser des phrases de longueurs variables.

Je n'irai pas plus loin ici dans les explications sur les structures de liste. Une partie du travail que vous aurez à mener consistera en effet à mener une étude documentaire, sur Internet, sur des livres de la B.U., pour vous documenter sur la manipulation de listes en général et en Prolog en particulier. Ce n'est qu'une fois que vous aurez acquis une certaine compétence sur le sujet (compétence qui vous sera utile au semestre 2), que vous pourrez aborder la partie réalisation du projet, qui est décrite ci-après. Dans l'immédiat, quelques pointeurs sur les listes en Prolog :

- **Internet**
  - Jean-Yves Antoine : www.info.univ-tours.fr/~antoine/documents_enseignement/LOGIQUE_CM_LP1_Prolog.pdf
  - Isabelle Tellier (prédicats de base) : www.lattice.cnrs.fr/sites/itellier/Prolog/corriges.php?td=3
  - Learnprolog (anglais) : www.learnprolognow.org/slides/official/LPNchapter4.pdf
- **Ouvrages disponibles à la bibliothèque universitaire de Blois** (cote 005.133 PRO)
  - Louis Gacôgne (2009) Prolog : programmation par l'exemple, Hermann : Paris.
  - Ivan Bratko (2001) Prolog : programming for artifical intelligence, Pearson/Addison Wesley.
  - William Cloksin, Christopher Mellish (1994) Programming in Prolog. Springer-Verlag, Berlin.

Le dernier ouvrage, en anglais, est une reference et certainement le plus exhaustif. Le premier a pour lui le fait d'être rédigé en français. La BU de Tours dispose par ailleurs de nombreux ouvrages sur le sujet, que vous pouvez faire venir à la BU de Blois, dont une autre référence qui a eu la bonne idée d'être traduite en français :

- Leon Sertling, Ekud Shapiro (1990) L'art de Prolog, Masson:Paris.

Avec ces informations, vous devez pouvoir aborder le sujet à proprement parler du projet.

3. Sujet : un problème schtroumpfement schtroumpf

Craignant de voir sa côte de popularité chuter comme celle de son premier ministre, François Hollande cherche depuis quelques mois à s'impliquer au minimum dans les affaires de politique intérieure afin d'apparaître comme un Président à la stature internationale au-dessus des problèmes franco-français. Après avoir initié une intervention militaire au Mali effectivement bien maitrisée, il lance désormais les troupes françaises dans une action d'aide au maintien de la paix en République Centrafricaine. Mais déjà cherche-t-il à avancer un nouveau pion sur cet échiquier géostratégique. Aussi saute-t-il de joie lorsqu'il apprend qu'il est invité en visite d'Etat au pays des schtroumpfs ! Malheureusement, le chef du protocole est incapable de trouver un traducteur parlant le langage schtroumpf. Les services de la présidence font alors appel à vos talents de programmeur pour réaliser un petit traducteur automatique français/schtroumpf. C'est la réalisation de ce traducteur qui est l'objet du projet.

Représentation des connaissances — On cherche à réaliser un programme Prolog qui traduise une phrase écrite en français en une phrase schtroumpf. Pour cela, on représentera tout énoncé par une liste de mots, i.e. une liste d'éléments de type chaîne de caractères. Par exemple :

Vive le peuple schtroumpf se représenté par `[vive,le,peuple,schtroumpf]`

Pour représenter les mots des dictionnaires français et schtroumpf, on souhaite par ailleurs utiliser les prédicats suivants :

- `verbe(X,TPS)` est vrai si X est un verbe conjugué au temps TPS.
- `adverbe(X)` est vrai si X est un adverbe.
- `nom(X)` est vrai si X est un nom commun.
- `autre(X)` est vrai pour tous les autres types de mots.

Vocabulaire – On considère à titre d'exemple le vocabulaire restreint suivant : `{je, vous, la, au, de, un, nom, chaleureusement, bienvenue, souhaite, france, suis, president, république, normal, reviendrai, serai }`.

1. Représenter ce vocabulaire dans un programme Prolog à l'aide des prédicats définis précédemment.

Traduction simple – En première approximation, le langage schtroumpf est très proche du français. Pour effectuer la traduction, il suffit en effet de remplacer tous les verbes français par le mot *schtroumpfe* et tous les adverbes par *schtroumpfement*. Par exemple :

Reste ici, je reviens vite est traduit par *Schtroumpfe ici, je schtroumfe schtroumpfement*

2. Écrire un prédicat d'arité 2 `trad_mot(FR,S)` qui réussit si S correspond à la traduction en schtroumpf du mot français FR à la traduction en schtroumpf d'un énoncé français représenté par la liste L_FR. On pensera pour cela à faire attention à la redoutable conjugaison des verbes schtroumpfs. Compte-tenu des difficultés inhérentes à la conjugaison schtroumpf, on se contentera d'étudier les verbes à la première personne du singulier au présent et au futur :

- présent *schtroumpfe*
- futur simple *schtroumpferai*

Par exemple, on aura :

```
?- trad_mot(suis,S)
YES S = schtroumpf
```

Par exemple, on aura :

```
?- trad_sple([je,suis,normal],S)
YES S = [je,schtroumpf,normal]
```

3. En utilisant le prédicat précédent, écrire alors un prédicat d'arité 2 `trad_sple(L_FR,L_S)` qui réussit si la liste L_S correspond à la traduction en schtroumpf de la phrase française FR représenté par la liste L_FR.

Traduction complexe – En réalité, le langage schtroumpf est d'une subtilité déroutante. En effet les noms composés du type `N de N ou N de la N` (*pomme de terre*, *livre de compte*, *président de la république*) ne conservent pas leur forme française mais se traduisent au contraire à l'aide du vocable *schtroumpf.* La difficulté ici étant que la traduction diffère sur l'on parle le schtroumpf du Nord ou le schtroumpf du Sud. On a en effet la règle de traduction suivante :

- N de N français : *président de la république*
- Schtroumpf du Nord *schtroumpf de la république*
- Schtroumpf du Sud *président de la schtroumpf*

4. En utilisant le prédicat précédent `trad_sple` pour déjà réaliser une première traduction simple, écrire un prédicat d'arité 2 `trad_complexe(L_FR,L_S,L)` qui réussit si la liste L_S correspond à la traduction complète dans la langue L de la phrase française FR représenté par la liste L_FR. L étant égale à `sud` pour le schtroumpf du Sud et `nord` pour le schtroumpf du Nord.

Par exemple, on aura :

```
?- trad_complexe([je,suis,le,president,de,la,republique],S,sud)
YES S = [je,schtroumpfe,le,president,de,la,schtroumpf]
```

Affichage – Enfin, il serait sympathique d'avoir un affichage plus lisible des énoncés traduits pour faciliter le travail de notre pauvre président.

5. Écrire enfin un prédicat d'arité 2 `affiche(L)` qui affiche sous forme de phrase le contenue d'une liste L.

Par exemple, on aura :

```
?- affiche([je,suis,le,president,de,la,republique])
YES je suis le president de la republique.
```

6. Ecrire enfin un prédicat d'arité 2 `trad_final(PH,Lgue)` qui affiche la traduction dans la langue schtroumpf `Lgue` de la phrase française `PH.On aura par exemple :`

```
?- trad_final([je,suis,le,president,de,la,republique],sud)
YES la traduction est : je schtroumpfe le president de la schtroumpf
```

4. Travail à rendre

Vous rendrez votre proposition de programme, dument testée auparavant, en déposant un fichier `.pl` sur le site consacré à cet enseignement sur votre environnement numérique de travail. Votre fichier suivant la règle de dénomination suivante : `schtroumpf_prenom_nom.pl`.

Le dépôt des projets est autorisé jusqu'au vendredi 10 janvier 2013 à 12 :00. Mais bien entendu, vous aurez tout intérêt (bonus) à proposer une solution à une date antérieure. Que le meilleur schtroumpfe et que la schtroumpf soit avec vous !